Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Technology 6 (2012) 315 – 323

Procedia Technology

2nd International Conference on Communication, Computing & Security [ICCCS-2012]

Binarization Based Image Edge Detection Using Bacterial Foraging Algorithm

Om Prakash Verma[a,*], Rishabh Sharma[b], Deepak Kumar[c]

[a]*Department of Information Technology, Delhi Technological University, Delhi, India*
[b]*Ericsson India Global Services Pvt. Ltd, Software Engineer, Noida, UP, India*
[c]*National Informatics Center, Delhi State, Ministry Of Communication And IT, Govt. Of India, Delhi, India*

Abstract

Bacterial Foraging Algorithm (BFA) is one of the powerful bio-inspired optimization algorithms which attempt to imitate the single and groups of *E. Coli* bacteria. In BFA algorithm, a set of bacteria try to forage towards a nutrient rich medium to get more nutrients. In this scheme, an objective function is posed as the effort or a cost incurred by the bacteria in search of food. In the present, an approach is presented for edge detection in a binarized image using bacterial foraging algorithm. .First binarization is applied to the input image to get an image matrix consisting of only the intensity values 0 and 255 of 8-bit image and then a swarm of bacteria are enthrusted on the binary image for extraction of edge information. Edges are detected by calculating the difference between intensity values of the present pixel with each of the neighboring eight pixels. Whenever the bacteria finds this intensity difference of 255 it will treat that pixel as its food and mark it as an edge pixel.

Keywords: Bacterial foraging algorithm; Edge; optimization; Binarization and heuristic.

1. Introduction

Edge detection is a terminology in image processing and computer vision, particularly in the areas of feature

* Corresponding author. Tel.: +0-991-005-0177.
E-mail address: opverma@dce.ac.in

doi:10.1016/j.protcy.2012.10.038

detection and feature extraction, to refer to algorithms which aim at identifying points in a digital image at which the image brightness changes sharply or more formally has discontinuities. Edge detection is a fundamental of low-level image processing and good edges are necessary for higher level of image processing [1].The edges provide important visual information since they correspond to major physical, photometrical or geometrical variations in scene object.

Edges include the most important information in the image, and can provide the information of the object's position [2]. Edges extracted from non-trivial images are often hampered by fragmentation, meaning that the edge curves are not connected, missing edge segments as well as false edges not corresponding to interesting phenomena in the image – thus complicating the subsequent task of interpreting the image data.

Traditional approaches to edge detection consists of edge detectors by Sobel [3], Prewitt [4], Kirsch [5], Frei-Chen [6], Canny [7] etc. Sobel edge operator, the Prewitt edge operator and the Robert edge operator used first-order derivative operators. In traditional approaches, each set of operation are applied for each pixels which leads to the creation of computationally expensive algorithms.

Over the last decades, various swarm intelligence based approaches have been adopted by researches in order to solve complex computational problems such as edge detection. Particle Swarm Optimization (PSO) [8] employs a swarming in which the movements of the particles are guided by the swarm's local best position as well as global best position in the required search-space. Recently Verma *et al*. [9] proposed a new optimal approach for edge detection using universal law of gravity and ant colony optimization. In this approach, the theory of universal gravity is used to calculate the heuristic function which guides the ant towards edge pixels. In 2007, Genyun Suna *et al* [10] have introduced an edge detection algorithm based on the law of universal gravity. In this approach, every image point is assumed as a celestial body, which has relationships with other neighboring image points. Recently Verma *et al*.[11] have developed an algorithm for edge detection using BFA in which direction of movement of bacteria is found using a directional probability matrix derived from ant colony optimization (ACO).

In 2002, another evolutionary bio-inspired algorithm was introduced by Passino *et al*. [12] known as bacterial foraging algorithm (BFA). BFA is the offshoot of the behavior of some species of bacteria like E. coli. There are four stages in the life cycle of bacteria namely: 1) Chemotaxis, 2) Swarming, 3) Reproduction and 4) Elimination and Dispersal. These stages in the search space generate an optimum solution to the problem of optimization.

Binarization is a process of converting a grey scale image into a black and white image. The process is often achieved by global or local thresholding. Binarization decreases the computational cost of subsequent processing compared to grey level image information.

In the proposed approach the combination of binarization and bacterial foraging algorithm is used to tackle the edge detection problem. Binarization is used to convert the input image into binary image and bacteria goes under foraging for detecting edges after being randomly placed on the binary image.

The rest of the paper is organized as the follows. The BFA is briefly reviewed in Section 2. Then, the algorithm of the proposed edge detector is presented in Section 3. The experimental results are given in Section 4 and conclusions are presented in Section 5.

2. Bacteria Foraging Technique

Bacterial foraging algorithm attempts to model the individual and group behaviour of *E.Coli* bacteria as a distributed optimization process. Foraging can be modelled as an optimization process where bacteria seek to maximize the energy obtained per unit time spent during foraging

Since its inception, BFA has been finding many important applications in real-world optimization problems from diverse domains of science and engineering. One key step in BFA is the computational chemotaxis, where a bacterium (which models a candidate solution of the optimization problem) takes steps over the foraging landscape in order to reach regions with high-nutrient content (corresponding to higher fitness).

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an

E.coli bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently, leading to tumbling of bacteria. In a rich medium the bacterium tumbles less number of times, whereas in a harmful place, it tumbles frequently to find a nutrient gradient. Moving the flagella in the counter clockwise direction helps the bacterium to swim at a very fast rate.

BFA mimics the four principal mechanisms observed in a real bacterial system namely; A) chemotaxis, B) swarming, C) reproduction, and D) elimination–dispersal.

A. Chemotaxis

Chemotaxis is a very important step in the bacterial foraging process. The direction of movement of bacterium is decided depending upon the rotation of the flagella. Each bacterium decides whether it should swim (move in a predefined direction) or tumble (move in an altogether different direction).

This process simulates the movement of an *E.coli* cell through swimming and tumbling via flagella. Suppose $\theta^i(j, k, l)$ represents the i^{th} bacterium at j^{th} chemotactic, k^{th} reproductive, and l^{th} elimination–dispersal step. *C(i)* indicates the size of the step taken in the random direction specified by the tumble (run length unit). Then, in computational chemotaxis, the movement of the bacterium may be represented by

$$\theta^i(j+1,k,l)=\theta^i(j,k,l)+C(i)\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}} \tag{1}$$

where $\Delta(i)$ indicates a random number in R^2 and *C(i)* indicates the length of step size.

B. Swarming

In this step, the bacterium that has searched an optimum path, signals other bacteria so that they can together reach the desired optimum path swiftly. As each bacterium moves, it releases an attractant to signal other bacteria to swarm towards it. Also, each bacterium releases a repellent to warn other bacteria by keeping a safe distance from them.

The cell-to-cell signaling in *E. coli* swarm can be represented by the following function:

$$\begin{aligned} &J_{cc}(\theta,P(j,k,l)) \\ &\quad=\sum_{i=1}^{S}J_{cc}(\theta,\theta^i(j,k,l)) \\ &\quad=\sum_{i=1}^{S}\left[-d_{attractant}\exp\left(-w_{attractant}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2\right)\right] \\ &\quad+\sum_{i=1}^{S}\left[h_{repellant}\exp\left(-w_{repellant}\sum_{m=1}^{p}(\theta_m-\theta_m^i)^2\right)\right] \end{aligned} \tag{2}$$

Where $J_{cc}(\theta,P(j,k,l))$ is the objective-function value to be added to the actual objective function (to be minimized) to present a time-varying objective function.

The coefficients $d_{attractant}$, $w_{attractant}$, $h_{repellant}$, *and* $w_{repellant}$ control the strength of the cell-to-cell signaling.

C. Reproduction

After N_c chemotactic steps, a reproduction step is taken. Let N_{re} be the number of reproduction steps to be taken. Let *S* is assumed to be a positive even integer such that

$$S_r=\frac{S}{2} \tag{3}$$

be the number of population members who have had sufficient nutrients so that they will reproduce (split in two) with no mutations.

D. Elimination Dispersal

This step regulates the population of bacteria by eliminating the bacteria that land in low nutrient or noxious regions. Dispersal on the other hand locates the bacteria in new regions, which might remain unexplored with limited number of bacteria taken. Let N_{ed} be the number of elimination-dispersal events, and for each elimination-dispersal event each bacterium in the population is subjected to elimination-dispersal with probability P_{ed} .

3. The proposed image edge detection approach

In this paper, bacterial foraging technique is used as the basis of detecting edges of the input image. Edges are detected by first applying a global threshold to the input image to form a binary image. Then a group of bacteria are placed at random position on the binary image and chemotactic steps of swimming and tumbling are performed. Bacteria's foraging behavior is driven by the difference in the intensities of the neighboring pixels. Appropriate edge pixels are found by calculating the difference between intensity values of a central pixel of interest with each of its eight neighboring pixels.

Steps of the proposed approach:

A. Image binarization

The binarization process involves the assignation of pixels to either foreground or background objects by comparing their intensity values to some prescribed or automatically selected thresholds. Thresholds are applied either globally or locally. In case of global threshold, all pixels above a defined value are set to white and all the pixels below it are set to black. It is difficult to find a global threshold value for all images hence it is often calculated experimentally. In the proposed approach, binarization process is carried out by applying a global threshold on an input image so that the image matrix consists of only values 0 and 255 (Fig 1). In present case, the threshold value selected is 127.

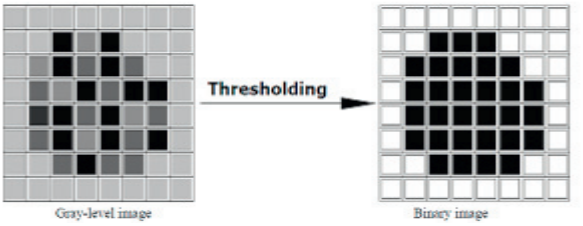

Fig 1 Conversion of gray scale image into binary

B. Edge detection

a. Initialization and search space identification

The bacteria are placed randomly all over the image which would then undergo foraging in search for food.

b. Initialize parameters

Let p be the dimension of the search space ,S be the number of bacteria placed on the image, N_c be the Number of chemotactic steps, N_{re} be the number of reproduction steps, N_{ed} be the number of elimination-dispersal steps, P_{ed} be the elimination-dispersal with probability, N_s be the Swim length and r be run-length of each bacterium.

c. Chemotactic step

i. Compute fitness function: $J(b,a,k,l)$

Fitness function of a bacterium b in its a^{th} chemotactic, k^{th} reproductive, and l^{th} elimination–dispersal step at position *(x, y)* is equal to the intensity value of the pixel at position *(x, y)*. The intensity value of a pixel will be either 0 or 255 because of binarization (Step 1).

ii. Compute cell-to-cell attractant–repellant profile to simulate the swarming behavior:

$$J(b,a,k,l)=J(b,a,k,l)+J_{cc}(\theta^b(a,k,l),\theta(a,k,l)) \quad (4)$$

Where $J(b,a,k,l)$ is the fitness function and $J_{cc}(\theta^b(a,k,l),\theta(a,k,l))$ is calculated using the Eq.2.

iii. Tumble:

A bacterium starts the process by tumbling. That is, it first moves in a random direction in search for food. The decision for this direction is done by calculating the difference between the intensity values of the present pixel (on which the bacterium is placed) with each of the eight neighboring pixels. In the above binarized image, this difference in intensity is 255 (Fig 1). Wherever the bacteria finds this intensity difference of 255, it will treat that pixel (black pixel having intensity 0) as its food and will replace this pixel with a pixel having intensity 127(it could be any random value between 0 and 255 except 0 and 255) (Fig 2).

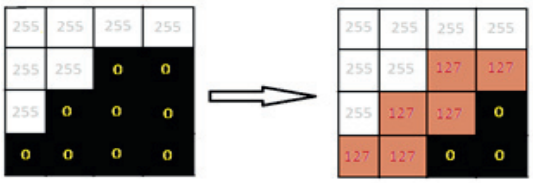

Fig 2 Allocating intensity of 127 wherever bacteria finds a difference of 255 in intensity of neighboring pixel.

These difference values are stored in a difference matrix to ease the task of finding out the pixel which would result in the maximum variation. Wherever there is an edge in an image, that corresponding pixel has a sharp increase in its value as compared to its eight neighboring pixels.

iv. Swim

The bacterium is moved to the new location using the Eq. (1). Since the bacteria aim at maximizing the energy per unit time, so they are always in search for patches with higher nutrient concentration and lesser noxious substances. It avoids noxious region and thus negative gradient region and prefers to move in positive nutrient gradient.

Thus, the bacterium follows the following criterion:

- Wherever it gets a positive nutrient gradient, it swims in that direction.
- And in neutral medium, it tumbles but if it reaches its maximum number of chemotactic steps and still not finds a positive nutrient gradient medium then it dies and it completely avoids negative nutrient gradient.

As a result, this pixel becomes the new location of the bacterium and the direction of motion of the bacterium is recorded.

d. Reproduction step

For given k and l, and each bacterium $i = 1, 2\ldots, S$

i. Sum

$$J^i_{health}=\sum_{j=1}^{Nc+1}J(b,a,k,l) \quad (5)$$

ii. Sort

Sort bacteria and in order of ascending cost J_{health}

iii. Split and Eliminate

The S_r bacteria with the highest J_{health} values die and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made and are placed at the same location as their parent).

e. Eliminate and dispersal step

In the elimination-dispersal step those bacterium which are eliminated are simply dispersed to a random location. Each bacterium is subjected to elimination-dispersal with probability P_{ed}.

The Algorithm and the pseudo code

[Step 1]: Convert the input image A into a binary image B.

[Step 2]: Initialize a new matrix C which is of the same size as A, and each pixel in the C matrix is assigned the value zero (i.e, black pixel).

[Step 3]: Initialize the parameters *p*, *S*, N_s , N_c , N_{re} , N_{ed} , P_{ed} , *r*, where:

p: Dimension of the search space

S: Number of bacteria placed on the image

N_c : Number of chemotactic steps

N_{re} : Number of reproduction steps

N_{ed} : Number of elimination-dispersal steps

P_{ed} : Elimination-dispersal with probability

N_s : Swim length

r: run-length of each bacterium

[Step 4]: Elimination-dispersal loop: *l = l + 1*

[Step 5]: Reproduction loop: *k = k + 1*

[Step 6]: Chemotactic loop: *a = a + 1*

[a] For *b=1,2,...S*, take a chemotactic step for each bacterium

[b] Compute fitness function, $J(b,a,k,l)$

Let, $J(b,a,k,l)=J(b,a,k,l)+J_{cc}(\theta^b(a,k,l),\theta(a,k,l))$ (i.e., add on the cell-to-cell attractant–repellant profile to simulate the swarming behavior)

[c] Let $J_{last}=J(b,a,k,l)$ to save this value, since we may find a better cost via a run.

[d] Tumble: The direction of the next move of the bacterium is found out by finding the difference of the current pixel with each of the eight neighboring pixels.

If the value of the current pixel is 255, and as soon as it is found out that there exists a neighboring pixel whose value is zero, then the value of this corresponding pixel in the C matrix is assigned the value 127.Similarly, if the value of the current pixel is zero and if there exists a neighboring pixel whose value is 255, then the current pixel *(i,j)* in matrix C is assigned the value 127.

In either case, the bacterium is moved to this new location and the direction of motion is recorded for swimming in that direction.

[e] Swim:

1) Let *m=0* (counter for swim length)

2) **While** ($m < N_s$)

Let *m = m+1*

3) **if** ($J(b,a+1,k,l) < J_{last}$)

Set $J_{last} = J(b, a+1, k, l)$

4) Let $\theta^b(a+1,k,l) = \theta^b(a,k,l) + C(b)\frac{\Delta(b)}{\sqrt{\Delta^T(b)\Delta(b)}}$

And use $\theta^b(a+1,k,l)$ to compute the next $J(b, a+1, k, l)$
Else let $m = N_s$
5) **End While** Loop
This results is a step of size *C(b)* in the direction of the tumble for bacterium *b*.
If j< N_c go to [Step 6][e] as the life of the bacteria is not over

[Step 7]: Reproduction:
[a] For the given *k* and *l*, and for each *i* = 1, 2, . . . , *S*, let $J^i_{health} = \sum_{j=1}^{Nc+1} J(b,a,k,l)$
be the health of the bacterium *i* (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters *C(b)* in order of ascending cost J_{health} (higher cost means lower health).
[b] The bacteria with the highest J_{health} values die, and the remaining S_r bacteria with the best values split (this process is performed by the copies that are made are placed at the same location as their parent).

[Step 8]: If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.

[Step 9]: Elimination–dispersal: For *i = 1, 2, . . . , S* with
Probability P_{ed}, eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If *l* < N_{ed},
then go to step 2;otherwise, end.

4. Experimental Results

The Experiments are conducted to evaluate the performance of the proposed approach using two test images, Camera and Einstein (Fig 3 and 4) and the performance is evaluated by comparing the edge map with conventional approaches.

Furthermore, various parameters of the proposed approach are set as follows:

$$\mu = 1, S = 100, d_{att} = 0.1, w_{att} = 0.2, h_{rep} = 0.1,$$
$$w_{rep} = 10, N_s = 10, P_{ed} = 0.95, N_{ed} = 100$$
$$N_{re} = 1, N_c = 100$$

In these experiments, traditional edge detectors are executed by MATLAB toolbox. The codes for our method were also written in MATLAB. The path traversed by a bacterium represents the edge pixels.

The proposed approach leads to less computational cost during the edge detection process because of binarization as compared to traditional edge detectors as they use grey level image information. The proposed method gives excellent results and robustly draws edges even in complex images. By applying global thresholding during binarization there are some points in the background image that are considered to be foreground and vice versa thus introducing noise in the binary image. The proposed approach is not able to differentiate between edge pixels and noise. However, the results also depend upon the kind of image one is

considering for edge detection. But overall the connectivity and number of pixels is high in comparison to traditional edge detectors.

The proposed approach tends to find meaningful edges in most images and is successful in optimally solving the edge detection problem by taking into consideration both the space and the time complexity.

5. Conclusions

In this paper, a new BFO based image edge detection approach has been successfully developed. Edge detection is a critical topic in computer vision and image processing. In the proposed approach the edge detection problem is handled by first converting the input image into binary image and then running the modified bacterial foraging algorithm on that image. Input image is first converted to binary image by applying a global threshold on the image and then bacteria are distributed randomly on the binary image for foraging. Edges are detected by bacteria by calculating the difference between intensity value of the present pixel with each of the neighboring eight pixel.

References

[1] M.B. Ahmad and T.S. Choi , "Local Threshold and Boolean Function Based Edge Detection", IEEE Transactions on Consumer Electronics, Vol. 45, No 3,pp. 674 -679, August 1999.

[2] RongWang, Li-qun gao, Shu Yang, Yan-Chun Liu, "An Edge detection method by combining fuzzy logic and neural networks", Machine Learning and Cybernetics,vol. 7,pp. 4539-4543,2005

[3] SOBEL, I., "Camera Models and Perception", Ph.D. thesis, Stanford University, Stanford, CA, 1970.

[4] J.M.S. Prewitt, "Object enhancement and extraction", in: B.S. Lipkin, A. Rosenfeld (Eds.), Picture Analysis and Psychopictorics, Academic Press, New York, NY, pp. 75-149, 1970.

[5] Kirsch, R., "Computer determination of the constituent structure of biological images", Computers and Biomedical Research, Vol. 4, pp. 315-328, 1971

[6] Frei and Chen, "Fast boundary detection: A generalization and a new algorithm", IEEE Trans. Computers, Vol. 26, pp. 988-998, 1977.

[7] J. Canny, "Computational Approach to Edge Detection", IEEE Transaction on Pattern Analysis and Machine intelligence, No. 6, pp. 679-698, 1986.

[8] Setayesh M., Zang M. and Jonhnton M., "A new homogeneity-based approach to edge detection using PSO", 24th International Conference Image and Vision Computing, New Zealand (IVCNZ), pp. 231-236, 2009.

[9] Verma O.P., Sharma R., "An optimal edge detection using universal law of gravity and Ant colony algorithm", World congress on communication and information technologies (WICT), 2011, in press

[10] Genyun Suna, Qinhuo Liua, Qiang Liua, Changyuan Jib, Xiaowen Lia ,"A novel approach for edge detection based on the theory of universal gravity" Pattern Recognition, Vol. 40(10), pp . 2766-2775, 2007

[11] Verma O.P., Hanmandlu M., Kumar P., and Jindal A, "A Novel Bacterial Foraging Technique for Edge Detection", Pattern Recognition Letters, No.8, pp.1187-1196, 2011.

[12] Passino,K. M., "Biomimicary of Bacterial Foraging for distributed optimization control", IEEE control system magazine, pp. 52-67,2002.

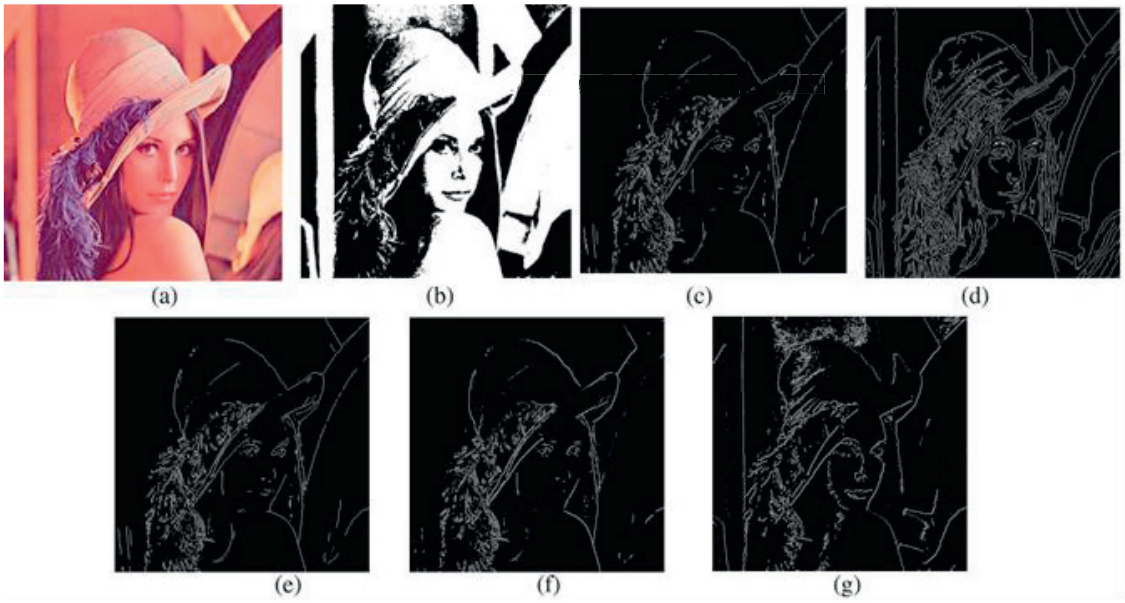

Fig. 3. (a) Original lena image (b)Binary image (c) Sobel Edge Detector (d) Canny Edge Detector (e) Prewitt Edge Detector (f) Robert Edge Detector (g) The proposed approach

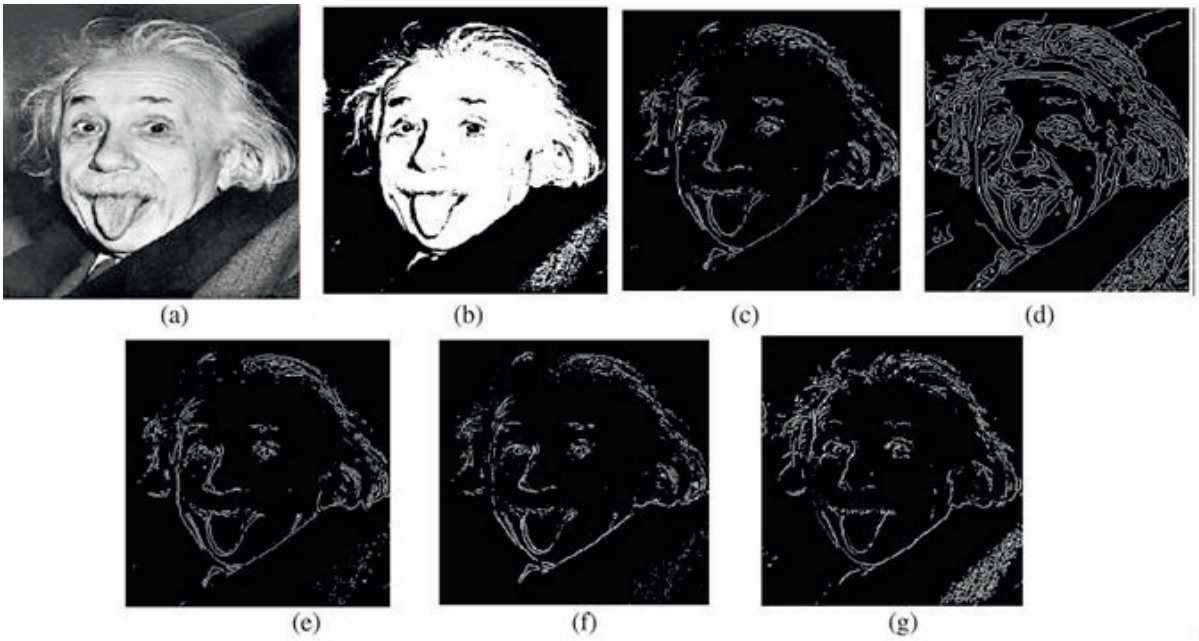

Fig. 4. (a) Original Einstein image (b)Binary image (c) Sobel Edge Detector (d) Canny Edge Detector (e) Prewitt Edge Detector (f) Robert Edge Detector (g) The proposed approach